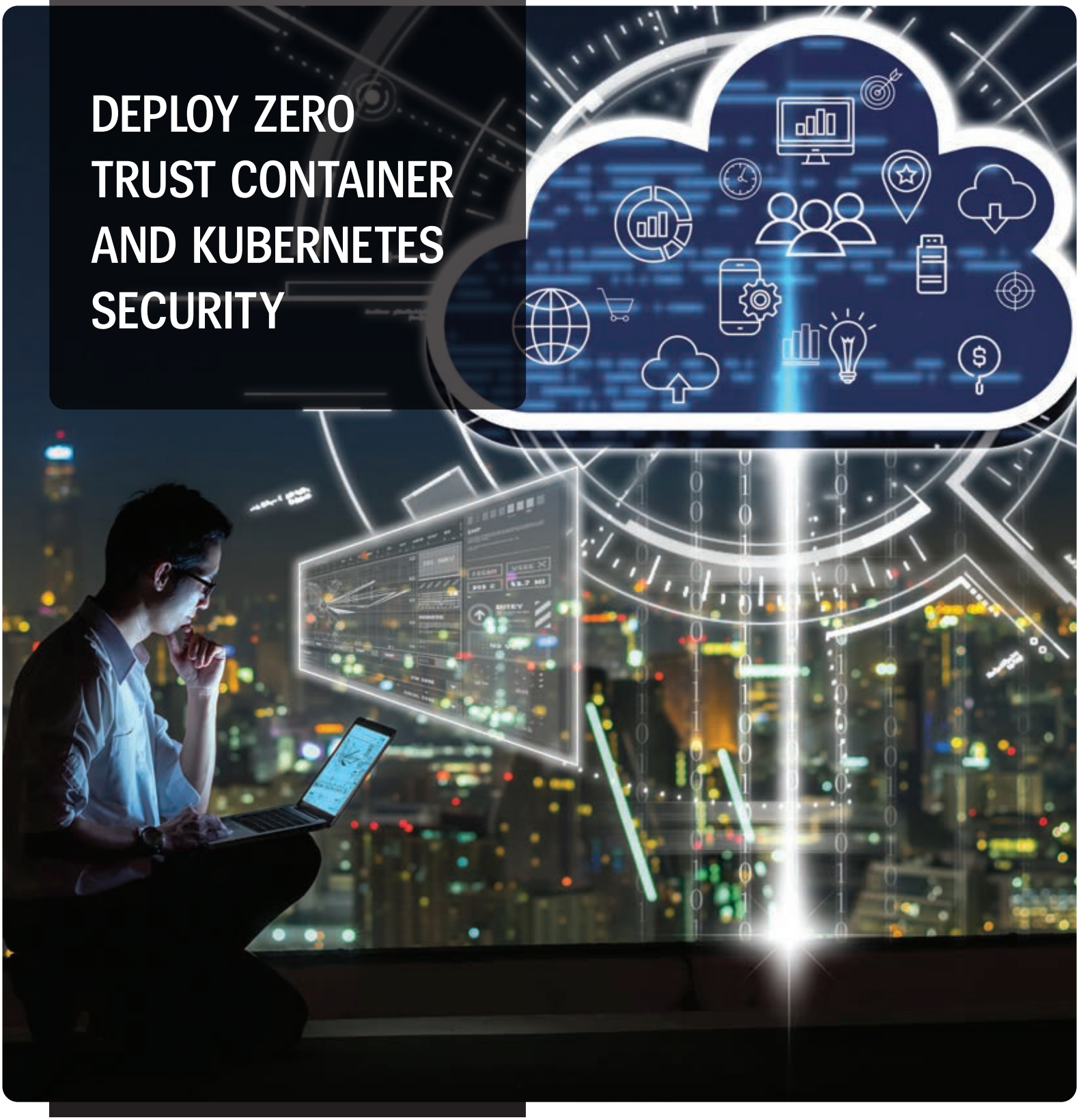
DEPLOY ZERO TRUST CONTAINER AND KUBERNETES SECURITY



STEALTH™

Gartner predicts that by 2022, more than 75% of global organizations will be running containerized applications in production - a significant increase from fewer than 30% today.

Though organizations can achieve many business benefits by using Kubernetes and Docker containers in tandem in IT environments, they do come with many vulnerabilities. According to a Forrester survey 43% of respondents indicated that security is a challenge hindering container adoption.

Unisys Stealth® with micro-segmentation, encryption and cloaking can help mitigate those vulnerabilities and further secure Kubernetes and container deployments.

Introduction

Docker containers and Kubernetes Orchestration improve software development and add much-needed agility to business systems. Applications can be deployed more quickly as system dependencies are handled outside the application code itself. Both of these technologies speed-up app development and conserve system resources.

Docker and Kubernetes are most frequently used together in large-scale production environments as complimentary products—Docker providing simple, straight forward deployment of container instances, and Kubernetes delivering automated scaling and management of large container deployments. Organizations can achieve the following business benefits by using Docker and Kubernetes in tandem in IT environments:

- Development Speed – Shorten development cycles by releasing new features/applications faster.
- Infrastructure Scalability – Scale infrastructure dynamically to meet business demands.
- Application Availability – Reduce downtime by deploying fixes faster.
- Cost Reduction – Decrease licensing costs by reducing and eliminating virtual environment and operating system requirements.
- Portability – Run apps across multiple OS and cloud environments with little or no modification.
- Performance – Spin up and run containers and apps in a fraction of the time required using VM partitions.

Docker has deservedly seen an explosion of growth since its introduction in 2013. The technology nearly single-handedly launched the 'cloud native' movement, providing unparalleled application performance and flexibility for businesses around the world. According to recent studies, Docker now has:

- More than 2 million Docker developers today.
- Over 80 billion downloads of Docker.
- An estimated 5.5 million Docker applications.

These pre-made, readily available open-source applications make Docker containers a natural choice for many businesses, particularly when first utilizing container technology in their environments. Businesses can download, run, and quickly modify open-source applications to suit their needs, without the cost and constraints of VM and host OS licensing costs or their inherent complexities.

Similarly, Kubernetes, with its Pod architecture, can be a highly useful addition to Docker when many microservices need to be spun up in tandem. Each Pod can host multiple microservice containers. However, the single most relevant IT business differentiator between Kubernetes and Docker alone is Kubernetes' ability to run (and manage) containers across an entire cluster of compute resources rather than the single-node limitation of Docker.

As mentioned, part of the tremendous growth of Docker and Kubernetes is their flexibility and open-source community support. Many organizations "turbo-charge" their development cycles by relying on public domain and readily available Docker images and Kubernetes to run their applications. However, many popular container images may harbor security flaws, some of which have been recently exploited by hackers and phishing attacks.

> **"An analysis of the 1,000 most popular Docker containers uncovered a variety of security vulnerabilities, some of which are critical."**
>
> **– Tech Republic, June 2019.**[i]

Rise in Container Vulnerabilities

As more and more apps find their way into the marketplace (and into containers), inevitable security lapses and potential flaws come to light that have been present in deployment architectures and 'home grown' implementations.

In a January 2019 survey, 60% of surveyed IT professionals working with containers experienced at least one security flaw in their container deployments. 6% experienced up to 100 vulnerabilities.[ii] Many of these can be traced to either improper container use of the root or lax security in user access controls.

Methods for Curbing Container Vulnerabilities

While isolating applications in a container or Kubernetes Pod does add a layer of security to their deployment through inherent isolation, how containerized apps interact with container run-time modules and their OS controls are increasingly a concern. And many public domain images may not have been fully vetted and scanned for vulnerabilities. However, best practice precautions can cut the likelihood of Container and/or Kubernetes breaches and security failures.

Challenges for Container security include:

According to Docker documentation:

There are four major areas to consider when reviewing Docker security:

- The intrinsic security of the kernel and its support for namespaces and cgroups;
- The attack surface of the Docker daemon itself;
- Loopholes in the container configuration profile, either by default, or when customized by users;
- The "hardening" security features of the kernel and how they interact with containers;

Challenges for Kubernetes security include:

- As the number of containers and Pods proliferate, visibility into individual applications becomes problematic. Therefore, tight control of privileges and authentication of each running application within Pods is essential to reduce risk.
- As a part of this authentication, admins must ensure permissions are granted for spinning up only registered and scanned images.
- Endpoints can always be a point of vulnerability. Therefore, each endpoint must be authenticated.
- Admins should use and extend Kubernetes built in Role Based Access Controls—ensuring that only authorized endpoints/users have access.

In short, both Kubernetes and pure container environments should adhere to a strong, Zero Trust security architecture: every step in the container/pod creation and runtime should be subject to authentication before granting access.

Zero Trust Security for Containers and Kubernetes

Adopted by many security professionals, Zero Trust is a security model of maintaining strict access controls at all times and not trusting any users (or processes they initiate) by default—even if the user is already within the network perimeter. It can provide a solid foundation for Container and Kubernetes security by not just relying on network perimeter defenses but also segmenting and securing internal container and Kubernetes actions. Attackers—through container-induced flaws or user misconfiguration—can circumvent perimeter controls, but internal micro-segmentation and authentication of user traffic can ensure every user and user action is authenticated.

Zero Trust guiding attributes are:

Security persistence: consistent security policy regardless of resource location.

Security ubiquity: security is present inside and outside of network perimeter (and containers).

Elasticity: ability to quickly adapt and change in to accommodate cloud and container environments.

Unisys Stealth Zero Trust and Micro-segmentation (SDEM) for Containers and Kubernetes

Unisys Stealth for containers and Kubernetes incorporates our latest software-defined encrypted micro-segmentation (SDEM) technology to deliver on all Zero Trust attributes—persistent, ubiquitous, and elastic protection in demanding container environments.

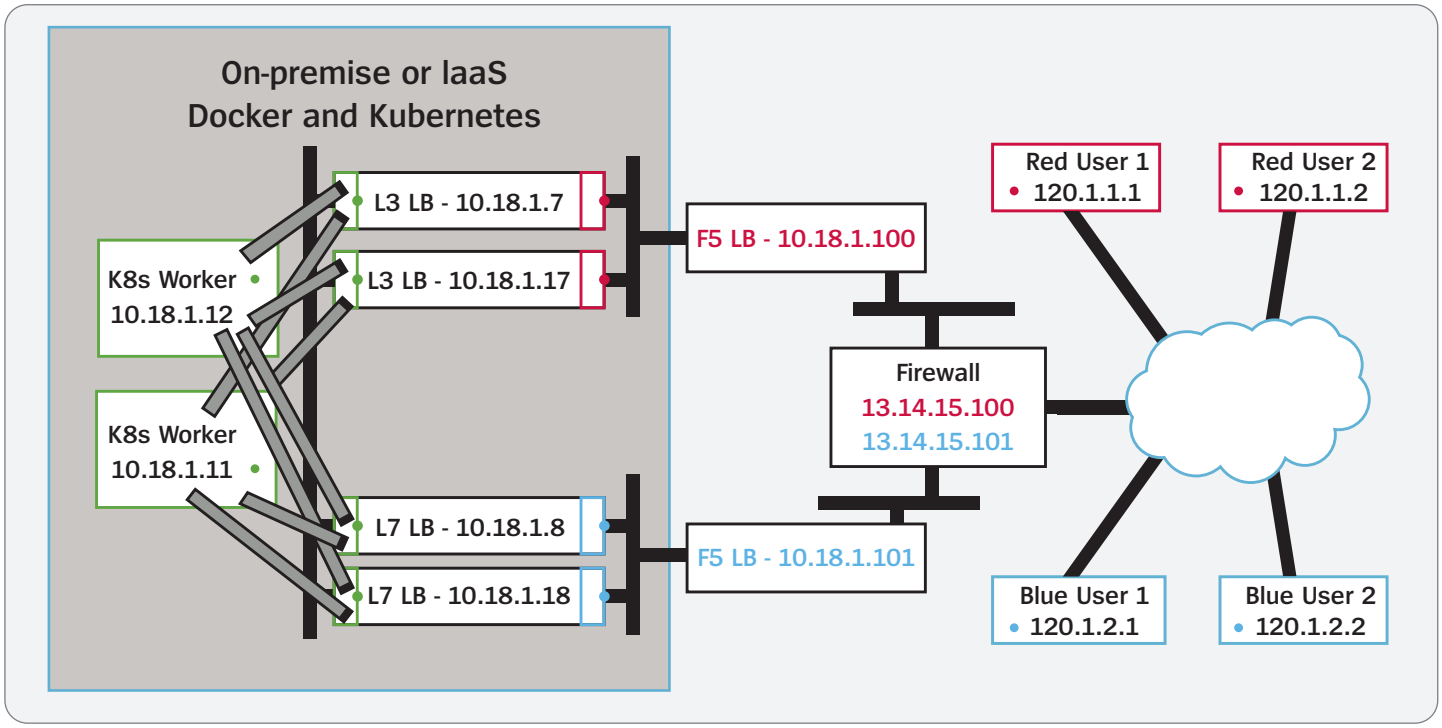


Figure 1. Stealth Zero Trust protection of Kubernetes Pods.

In typical Kubernetes cloud environments, lightweight Stealth™ agents at the user endpoints and within Kubernetes cloud load balancers ensure that only authorized and authenticated users have access to their allowed through its role-to-service mapping. Users are allowed access to and can see only the services they are assigned through their privileges. (See Figure 1.) For example, in Figure 1., the blue user has been assigned permissions to use 'Blue' services, regardless of location within the Kubernetes cluster, governed by its strict authentication at the endpoint through to the Stealth agent present in the Ingress nodes 10.18.1.7 and 10.18.1.17. All traffic to and from the endpoint is strongly encrypted. Similarly, the Red user is allowed access to only its permissible services and cannot see or use services that have not been assigned, even among services running on the same cluster.

In addition, all inter-node traffic inside the cluster is protected by Stealth (the green endpoints), so eavesdropping or packet injection on the cluster network itself is protected by strong encryption. For example, a misconfigured node cannot send or receive messages to or from the public network.

For added security, businesses can run Stealth(aware)™ which can aid in detecting 'misbehaving' applications or services and can assist in making sure whitelisted processes are authenticated and safe.

Unisys Stealth for Kubernetes provides:

- Stealth overlay that restricts users to only the Kubernetes service they are allowed to access, with masked IP addresses and paths to the cluster access point.
- Fully encrypted traffic from each access endpoint to its assigned Kubernetes cluster through the load balancer.
- Monitoring of access to and from the cluster(s).

Benefits include:

- Highly segmented security access controls without the need to configure firewalls or load balancers.
- Additional and complimentary security for Kubernetes RBAC.
- Encrypted traffic to and from Kubernetes clusters, regardless of network topology (cloud, on-prem., etc.).

For organizations using persistent Docker containers, Stealth enables stack-level controls for access to services running in static containers.

Unisys Stealth for container provides:

- An encryption overlay that creates separated micro-segments for the host and within containers —each user restricted to its assigned segment/path. The micro-segment restricts the hosts, ports, protocols and IP addresses the user (or endpoint) is permitted to communicate with.
- The control layer provides endpoints with access privileges for requested workloads based on identity and security policies.
- The data layer uses key-enabled point-to-point communication.

The benefits of Unisys Stealth for containers using Zero Trust SDEM:

- Cloaking of container resources making them undetectable from any unauthorized user, within or outside the network, or from other container(s).
- Encryption of all data in motion on host and within containers.
- Automated run-time deployment of all role-based security control policies.
- Dramatically reduced container attack surface, adhering to and improving on Docker best practices.
- Protect container resources by dynamically isolating threats.

By applying Zero Trust best practices and internal micro-segmentation, attack surfaces in containers or server run-time are dramatically reduced.

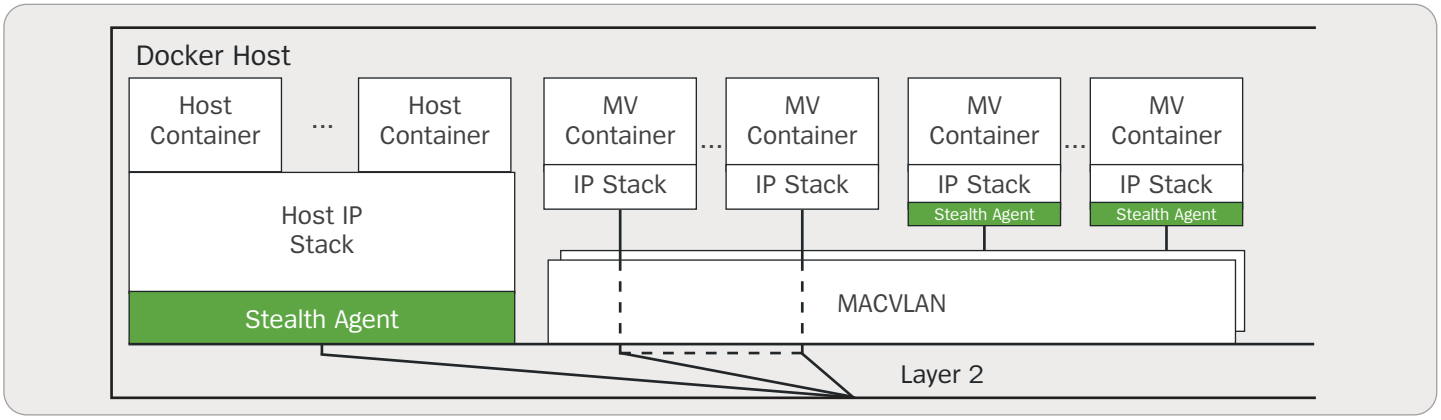


Figure 2. Stealth protection within Docker host and application container(s).

Summing up the Business Benefits

There is a marked increase in container technology as a flexible, cost-effective alternative to VMs. Whether organizations are deploying limited, static Docker containers and applications or deploying clusters of dynamic, on-demand Kubernetes, they should ensure their security posture is up to the task.

Unisys Stealth, with its Always On software-defined micro-segmentation *reduce the attack surface, minimize risks, dynamically isolate threats and help businesses realize the agility, availability, manageability, and scalability advantages offered by containerized microservices and Kubernetes*. This would also help in streamlining the toolsets used to secure application modernization and DevOps initiatives. And as container and cloud-native services and applications continue to evolve, Stealth's highly adaptive architecture will meet new challenges in the future.

---

[i] Tech Republic. "Docket Containers are Filled with Vulnerabilities." June 2019.
www.techrepublic.com/article/docker-containers-are-filled-with-vulnerabilities-heres-how-the-top-1000-fared/

[ii] ITPro Today. "Survey Shows Container Security Flaws Limit Adoption." January 2019.
www.itprotoday.com/containers/survey-indicates-container-security-concerns-limit-adoption

How to successfully use containers
techhq.com/2019/06/how-to-successfully-use-containers/

For more information or to request a demo, contact us

www.stealthsecurity.unisys.com/contact-us/

For more information visit www.unisys.com